Data Structures I: Stacks and Queues

UNIVERSIDAD EAFIT

Mauricio Toro
Department of Systems and Informatics
Universidad EAFIT

Cocktail of the day: Dry Martini

Disclaimer: Keep alcohol out of the hands of minors.

Cocktail of the day: Dry Martini

- 60ml gin
- 10ml vermouth
- 1 olive

Artificial Intelligence for chess uses trees

https://www.youtube.com/watch?v=NJarxpYyoFI

Complexity analysis

	insertFirst	insert(i)	deleteFirst	delete(i)	get(i)
ArrayList	O(n)	O(n)	O(n)	O(n)	O(1)
LinkedList	O(1)	O(n)	O(1)	O(n)	O(n)

Table: Complexity analysis of the operations of LinkedList and ArrayList.

Implementations of Lists

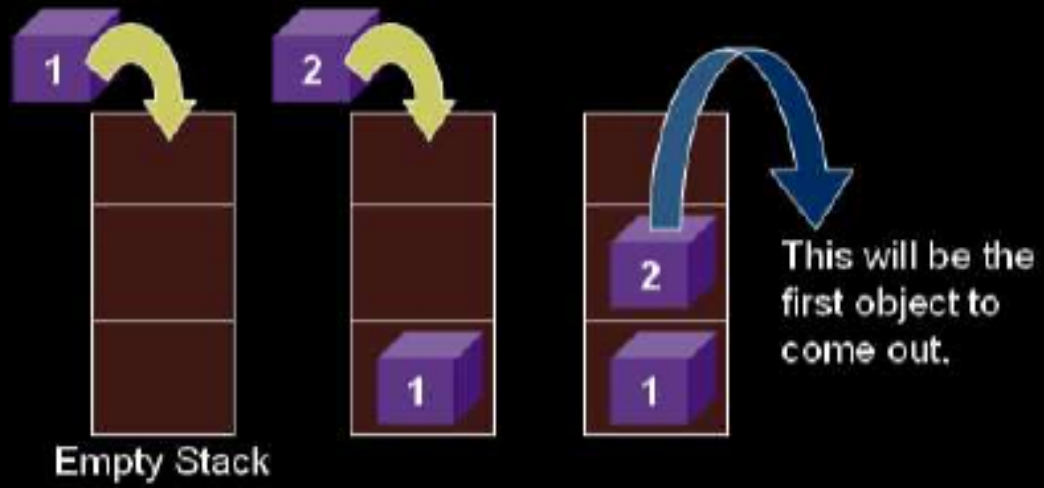
- Methods for both LinkedList and ArrayList are the same, but implementations are different.
- LinkedList is more efficient for insertion and ArrayList is more efficient for random access.

Implementations of Lists

- Methods for both LinkedList and ArrayList are the same, but implementations are different.
- LinkedList is more efficient for insertion and ArrayList is more efficient for random access.

Stacks

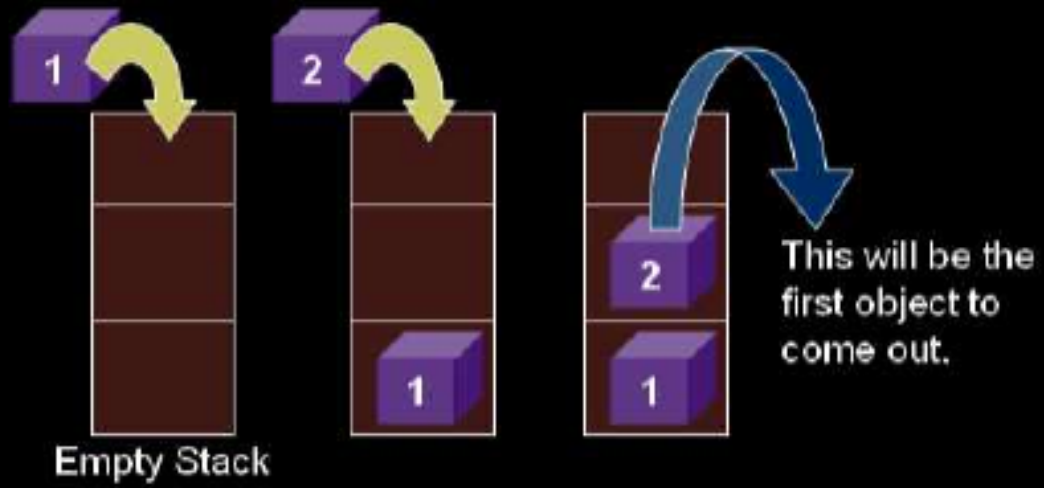
- A stack allows access to only the last item inserted.
- If you remove this item, then you can access the next-to-last item inserted



http://www.codeproject.com/KB/dotnet/Stacks/Stack.jpg

Stacks

- A stack allows access to only the last item inserted.
- If you remove this item, then you can access the next-to-last item inserted



http://www.codeproject.com/KB/dotnet/Stacks/Stack.jpg

Simulator of a Stack

http://visualgo.net/list.html

Application of Stacks

1. Check whether parentheses, braces, and brackets are balanced in a computer program source file.
2. Help traverse the nodes of a tree.
3. Search vertices of a graph
4. Most microprocessors use a stack-based architecture.
5. Some calculators use a stack-based architecture.

Application of Stacks

1. Check whether parentheses, braces, and brackets are balanced in a computer program source file.
2. Help traverse the nodes of a tree.
3. Search vertices of a graph
4. Most microprocessors use a stack-based architecture.
5. Some calculators use a stack-based architecture.

Application of Stacks

1. Check whether parentheses, braces, and brackets are balanced in a computer program source file.
2. Help traverse the nodes of a tree.
3. Search vertices of a graph
4. Most microprocessors use a stack-based architecture.
5. Some calculators use a stack-based architecture.

Application of Stacks

1. Check whether parentheses, braces, and brackets are balanced in a computer program source file.
2. Help traverse the nodes of a tree.
3. Search vertices of a graph
4. Most microprocessors use a stack-based architecture.
5. Some calculators use a stack-based architecture.

Application of Stacks

1. Check whether parentheses, braces, and brackets are balanced in a computer program source file.
2. Help traverse the nodes of a tree.
3. Search vertices of a graph
4. Most microprocessors use a stack-based architecture.
5. Some calculators use a stack-based architecture.

Operations of a Stack

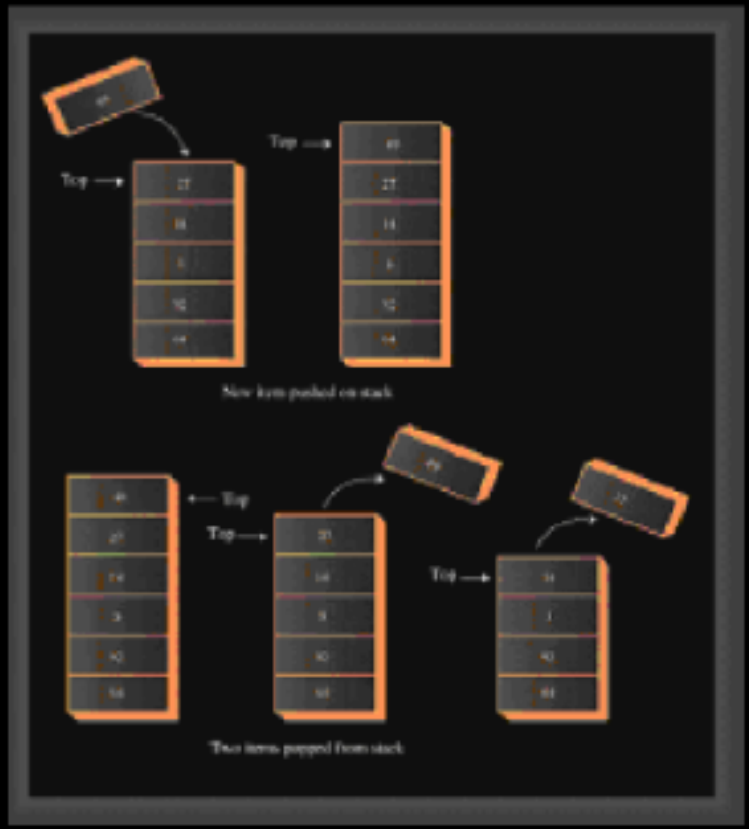


Figure: Operation of the StackX class methods. Taken from [Laf98].

Implementation of Stacks

```
class StackX
{
public StackX(int size);
public void push(int j);
public double pop();
public double peek();
public boolean isEmpty();
public boolean isFull();
}
```

Example: Reverse a String

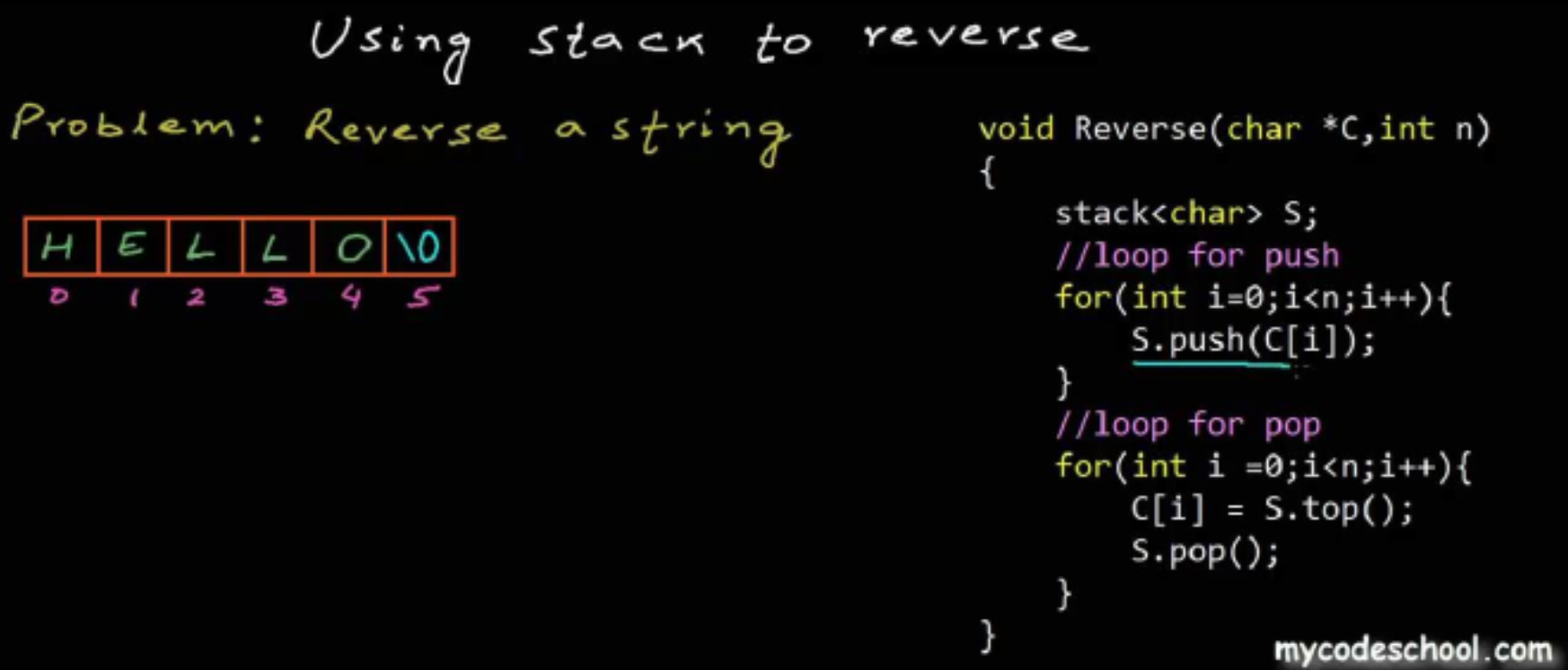


Figure: Reverse a string using a Stack. Taken from mycodeschool.com.

Example: Reverse a String (2)

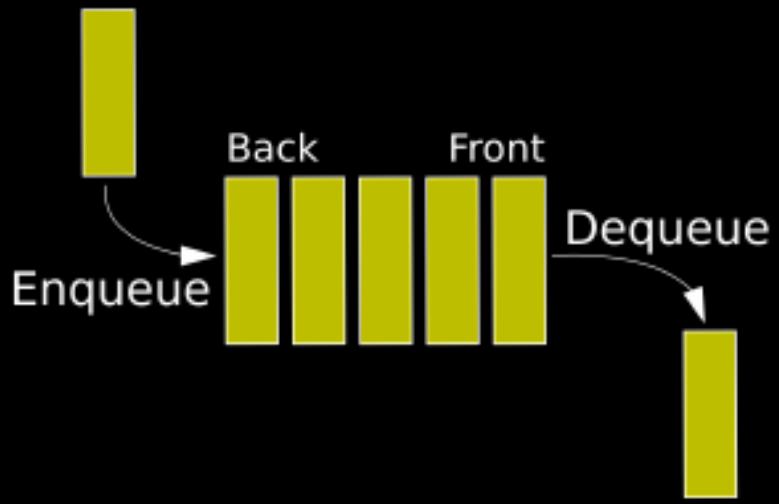
https://www.youtube.com/watch?v=hNP72JdOIgY

Example: Reverse a String

```
String str = "hola";
Stack<String> stack=new Stack<String>();
for (int i=0;i<str.length();i++)
  stack.push(str.substring(i,i+1));
String strrev = "";
while(!stack.isEmpty())
  strrev += stack.pop();
```

Queues

- In a queue, the first item inserted is the first to be removed



Taken from Wikipedia

Simulator of a Queue

http://visualgo.net/list.html

Application of Queues

- Breadth-First Search
- When a resource is shared among multiple consumers
  - Examples include CPU scheduling, Disk Scheduling...
- When data is transferred asynchronously between two processes
  - Examples include IO Buffers, pipes, file IO, TCP/IP...

Application of Queues

- Breadth-First Search
- When a resource is shared among multiple consumers
  - Examples include CPU scheduling, Disk Scheduling...
- When data is transferred asynchronously between two processes
  - Examples include IO Buffers, pipes, file IO, TCP/IP...

Application of Queues

- Breadth-First Search
- When a resource is shared among multiple consumers
  - Examples include CPU scheduling, Disk Scheduling...
- When data is transferred asynchronously between two processes
  - Examples include IO Buffers, pipes, file IO, TCP/IP...

Operations of a Queue

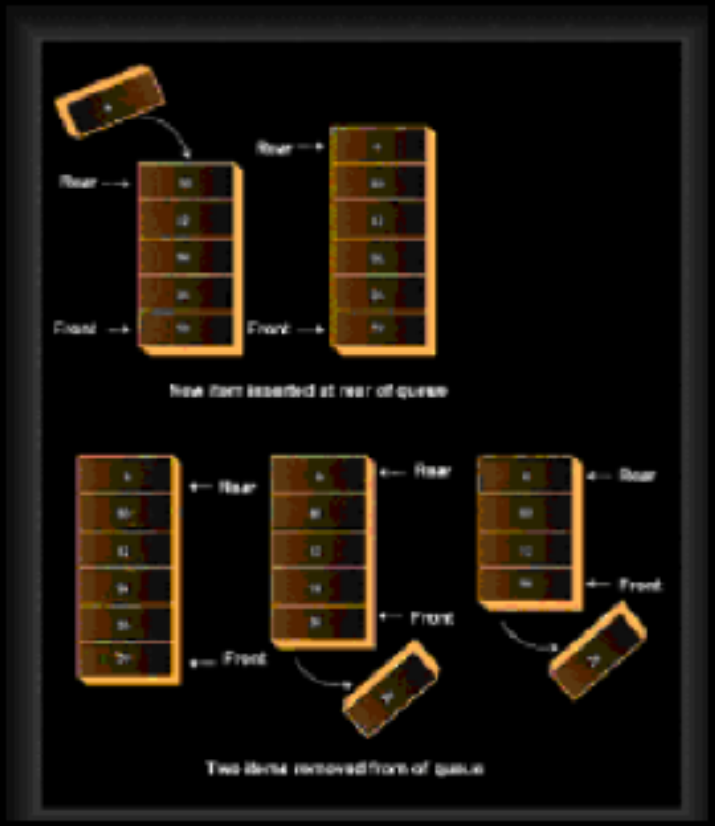


Figure: Operation of the Queue class methods. Taken from [Laf98].

Implementation of Queues

```
class Queue
{
public Queue(int s)
public void insert(int j)
public int remove()
public int peekFront()
public boolean isEmpty()
public boolean isFull()
public int size()
}
```

Queue operations example

```
Queue queue = new LinkedList();
//exception could be thrown
queue.add("item1");
//no exceptions thrown
queue.offer("Item3");
//Removes the first item
System.out.println(queue.remove());
//may returns false.
System.out.println(queue.poll());
//may return a null value
System.out.println(queue.peek());
```

Summary

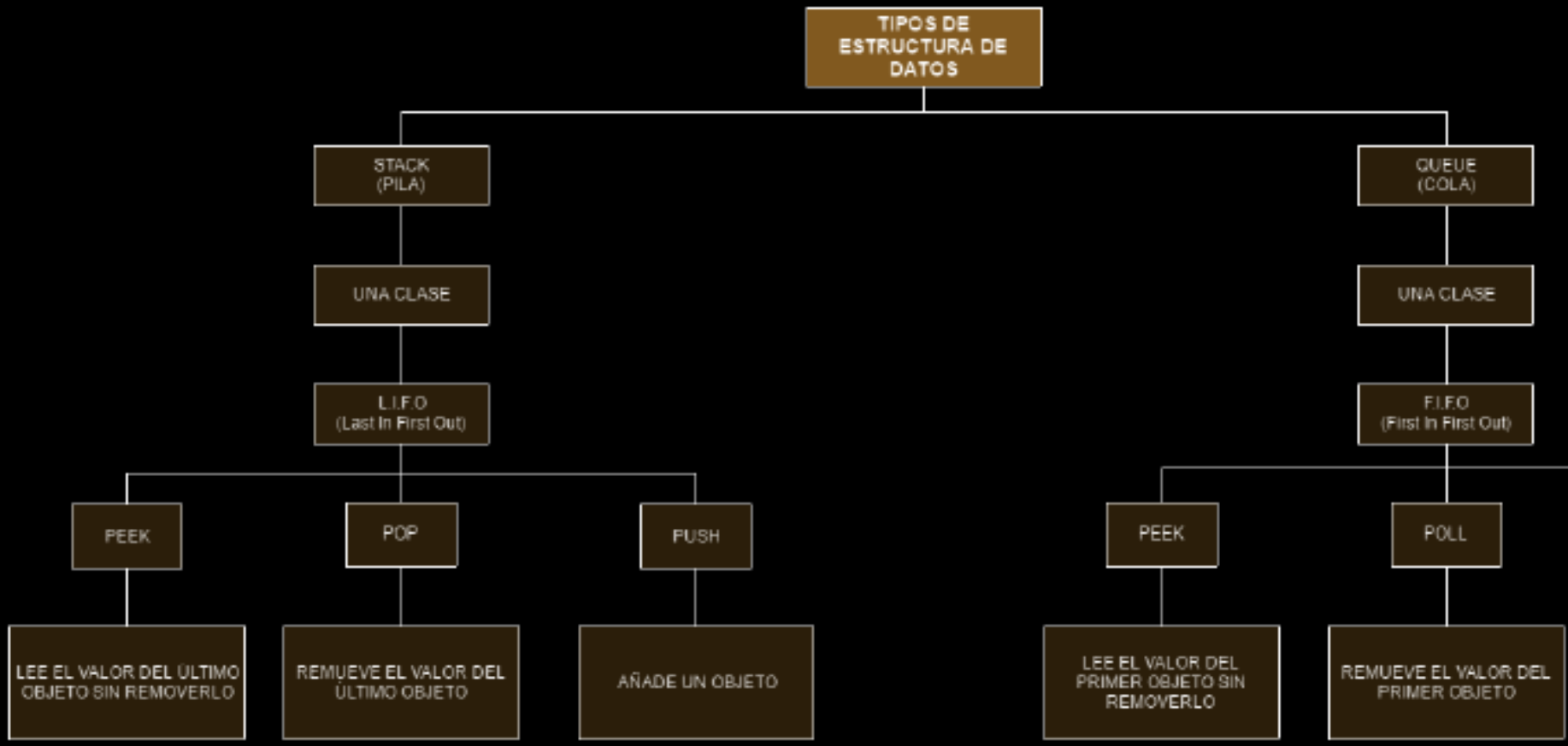


Figure: Mapa conceptual. Contribución de Camilo Villa.

Quiz questions

- A stack is also called "Last in, first out" (LIFO).
- A queue is also called "First in, first out" (FIFO).

References

- Please check the slides after class to learn how to reference images, trademarks, videos and fragments of code.
- Avoid plagiarism

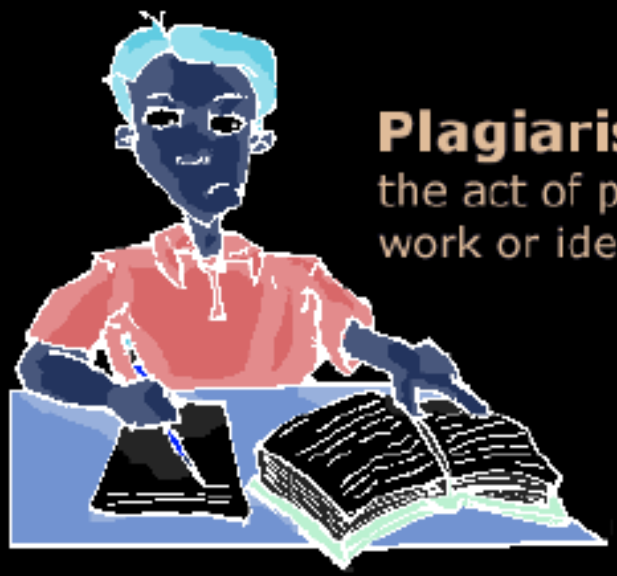


Figure: Figure about plagiarism, University of Malta [Uni09]

References

Robert Lafore.

Data Structures and Algorithms in Java.

QUE; 1 edition (8 November 2002), 1998.

University of Malta.

Plagarism — The act of presenting another's work or ideas as your own, 2009.

[Online; accessed 29-November-2013].

Further reading

- Stacks and Queues
  - Mitchell Waite. Data Structures and Algorithms in Java. Chapter 4.